



### Apple IIGS

### #96: Standard File Customization

Written by: Dan Strnad

November 1990

This Technical Note discusses particulars of using custom dialog boxes for the Open and Save File dialog boxes and custom drawing routines to display the files and folders listed.

---

#### About the Templates

Volume 3 of the *Apple IIGS Toolbox Reference* states the following about the Open File dialog box template for Standard File:

“The scroll bar item (item5) is not used for single-file calls. For multifile calls, this item contains the Accept Button Definition.”

What is not stated explicitly is that, although the scroll bar item is not used for single-file calls, a place holder for it must be included in the dialog box template. Another particular not explicitly stated is that the strings used by the item templates must be Pascal strings; no `listType` field is provided in the extended list control record as was present in the List Manager's original list record structure.

#### Custom Item Draw Procedures

Custom item draw procedures have the rectangle in which the item is to be drawn, the List Manager's `memRec` structure corresponding to that item, and a handle to the extended list control record available on the stack. By including a custom item draw procedure, programs are able to get a handle to the extended list control record. The custom item draw procedure could also make the handle available to other routines, such as the `dialogHook` routine. With the handle, programs can now perform specialized operations during a standard file call, such as checking which item is selected before allowing the user to cancel. The code fragment below (from DTS Apple II Sample Code #18, `AccessPriv`) illustrates the use of `SFPGetFile2` with a custom item draw routine.

```
static char SaveStr[] = "\pSave";
static char OpenStr[] = "\pOpen";
static char CloseStr[] = "\pClose";
static char DriveStr[] = "\pDrive";
static char CancelStr[] = "\pCancel";
```

```
static char FolderStr[] = "\pNew Folder";  
static char AcceptStr[] = "\pAccept";
```

```
ItemTemplate OpenBut640 = {1,
                           61,265,73,375,
                           buttonItem,
                           OpenStr,
                           0,
                           0,
                           0L};

ItemTemplate CloseBut640 = {2,
                            79,265,91,375,
                            buttonItem,
                            CloseStr,
                            0,
                            0,
                            0L};

ItemTemplate NextBut640 = {3,
                           25,265,37,375,
                           buttonItem,
                           DriveStr,
                           0,
                           0,
                           0L};

ItemTemplate CancelBut640 = {4,
                              97,265,109,375,
                              buttonItem,
                              CancelStr,
                              0,
                              0,
                              0L};

ItemTemplate Scroll640 = {5,
                          43,265,55,375,
                          buttonItem,
                          AcceptStr,
                          0,
                          0,
                          0L};

ItemTemplate Path640 = {6,
                        12,15,24,395,
                        userItem,
                        0L,
                        0,
                        0,
                        0L};

ItemTemplate Files640 = {7,
                         25,18,107,215,
                         userItem + itemDisable,
                         0L,
                         0,
                         0,
                         0L};

ItemTemplate Prompt640 = {8,
                           3,15,12,395,
                           statText + itemDisable,
                           0L,
                           0,
                           0,
                           0L};
```

```

/*****
*
* myDialogHook
*
*****/

pascal void myDialogHook(strip1,strip2)
long strip1;
long strip2;
{
}

/*****
*
* CustomItemDraw
*
*****/

pascal void CustomItemDraw(itemDrawPtr)
Pointer itemDrawPtr;
{
static unsigned int flag, dbr;          /* result, data bank register value */
byte      StringCount;
char      *ItemPascalString;
Word      ItemFileType;
Long      ItemAuxType;
Rect      *TheItemRectPtr;
MemRec    *TheMemRecPtr;
CtlRecHndl TheSFListControlHndl;
Point     MyOldPenPos,
          MyNewPenPos;

static char FileString[] = "xxxx yyyyyyyy ";

/* save our data bank and set current to global page */
dbr = SaveDB();
/* Get the Rect from High on the Stack */
TheItemRectPtr = (Rect *)(((long *)((long)&itemDrawPtr)+ 36L)));
/* save old pen position */
GetPen(&MyOldPenPos);
/* Set our pen position */
MyNewPenPos.h = TheItemRectPtr->h1 + 5;
MyNewPenPos.v = TheItemRectPtr->v2 -2;
MoveTo(MyNewPenPos);          /* relocate the pen */

/* get our member record; this is just to reveal where it is on the stack */
TheMemRecPtr = (MemRec *)(((long *)((long)&itemDrawPtr)+ 32L)));

/* get the list cntrol handle; ditto */
TheSFListControlHndl = (CtlRecHndl)(((long *)((long)&itemDrawPtr)+ 28L)));

StringCount = (byte) *itemDrawPtr;          /* get the string length */
ItemPascalString = itemDrawPtr;            /* set our user string */
ItemFileType = *(Word *) (itemDrawPtr+StringCount+1L); /* get our FileType */
ItemAuxType = *(Long *) (itemDrawPtr+StringCount+3L); /* get our AuxType */

/* format for display */
sprintf(FileString, "%.4x-%.8lx ",ItemFileType,ItemAuxType);
c2pstr(FileString);          /* turn it into a P string */
DrawString(FileString);      /* Draw it */
DrawString(ItemPascalString); /* catenate File name to the other
info */
FrameRect(TheItemRectPtr);
MoveTo(MyOldPenPos);          /* return the pen to starting
position */
RestoreDB(dbr);              /* restore our data bank */

```

```

}

/*****
*
* ChooseFolder
*
* presents user with dialog to select folder to show/set privileges of
*
*****/

void    SomeProc()
{
    DialogTemplate GetDialog640;

    GetDialog640.dtBoundsRect.v1 = 0;
    GetDialog640.dtBoundsRect.h1 = 0;
    GetDialog640.dtBoundsRect.v2 = 114;
    GetDialog640.dtBoundsRect.h2 = 400;
    GetDialog640.dtVisible = -1;
    GetDialog640.dtRefCon = 0L;
    GetDialog640.dtItemList[0] = &OpenBut640;
    GetDialog640.dtItemList[1] = &CloseBut640;
    GetDialog640.dtItemList[2] = &NextBut640;
    GetDialog640.dtItemList[3] = &CancelBut640;
    GetDialog640.dtItemList[4] = &Scroll640;
    GetDialog640.dtItemList[5] = &Path640;
    GetDialog640.dtItemList[6] = &Files640;
    GetDialog640.dtItemList[7] = &Prompt640;
    GetDialog640.dtItemList[8] = 0L;

    SFPGetFile2(    /* user selection of folder to get/set privs of */
        120, 53,
        CustomItemDraw,
        refIsPointer,
        prompt,
        0L,
        0L,
        &GetDialog640,
        myDialogHook,
        &myReply
    );
}

```

---

## Further Reference

- *Apple IIGS Toolbox Reference*, Volumes 1 & 3
- DTS Apple II Sample Code #18, AccessPriv